Dialog DataStar

options | logoff | feedback | help

databases | easy search

Advanced Search: INSPEC - 1969 to date (INZZ)

limit

Search history:

No.	Database	Search term	Info added since	Results	
7	INZZ	6 AND commit NEAR transaction$	unrestricted	18	show titles
8	INZZ	7 AND (rollback$ OR roll$ ADJ adj1 ADJ back$) NEAR transaction$	unrestricted	0	-
9	INZZ	7 AND (rollback$ OR roll$ ADJ adj1 ADJ back$)	unrestricted	0	-
10	INZZ	nest$ NEAR (database$ OR data ADJ adj1 ADJ base$ OR data ADJ adj1 ADJ bank$ OR transaction$)	unrestricted	416	show titles
11	INZZ	10 AND (rollback$ OR roll ADJ adj1 ADJ back$)	unrestricted	7	show titles
12	INZZ	11 AND commit	unrestricted	1	show titles
13	INZZ	rollback$ OR roll$ ADJ adj1 ADJ back$	unrestricted	1060	show titles
14	INZZ	13 AND commit	unrestricted	38	show titles
15	INZZ	14 AND (hierarch$ OR node$ OR level$ OR layer$ OR branch$)	unrestricted	12	show titles
16	INZZ	15 AND nest$	unrestricted	0	-

show all | hide | delete all search steps... | delete individual search steps...

Enter your search term(s): Search tips ☐ Thesaurus mapping

whole document

Information added since: or: none
(YYYYMMDD)

search

Select special search terms from the following list(s):

- Publication year
- Classification codes A: Physics, 0-1
- Classification codes A: Physics, 2-3
- Classification codes A: Physics, 4-5

10/055,208

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

document 1 of 3 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4886472, C9504-6160J-012; 950228.

Title
Locking in OODBMS client supporting **nested transactions.**

Author(s)
Daynes-L; Gruber-O; Valduriez-P; Ed. by Yu-P-S; Chen-A-L-P.

Author affiliation
Inst Nat de Recherche en Inf et Autom, Le Chesnay, France.

Source
Proceedings of the Eleventh International Conference on **Data** Engineering, Taipei, Taiwan, 6-10 March 1995.
Sponsors: IEEE Comput. Soc. Tech. Committee on **Data** Eng., Nat. Tsing Hua Univ., Providence Univ., Taiwan.
In: p.316-23, 1995.

ISSN
ISBN: 0-8186-6910-1, CCCC: 1063-6382/95/ ($4.00).

Publication year
1995.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
P Practical.

Abstract
Nested transactions facilitate the control of complex persistent applications by enabling both fine-tuning of the scope of **rollback** and safe **intra-transaction** parallelism. We are concerned with supporting concurrent **nested transactions** on client workstations of an OODBMS. Use of the traditional design and implementation of a lock manager results in a high CPU overhead: in-cache traversals of the 007 benchmark perform, at best, 4.5 times slower than the same traversal achieved in virtual memory by a nonpersistent programming language. We propose a new design and implementation of a lock manager which cuts that factor down to 1.8. This lock manager supports

nested transactions with both sibling and parent/child parallelisms, and provides object locking at a cost comparable to page locking. Object locking is therefore a better alternative due to its higher functionality. (8 refs).

Descriptors

cache-storage; client-server-systems; concurrency-control; distributed-databases; object-oriented-databases; software-performance-evaluation; transaction-processing; workstations.

Keywords

OODBMS client supporting **nested transactions;** object locking; complex persistent applications; **rollback;** safe intra **transaction** parallelism; concurrent **nested transactions;** client workstations; lock manager; in cache traversals; 007 benchmark; parent child parallelism; sibling parallelism.

Classification codes

C6160J (Object-oriented databases).
C6150N (Distributed systems software).
C6120 (File organisation).
C6160B (Distributed databases).

☑ **document 2 of 3** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

3965793, C91059234; 910829.

Title

ARIES/NT: a recovery method **based** on write-ahead logging for **nested transactions.**

Author(s)

Rothermel-K; Mohan-C; Ed. by Apers-P-M-G; Wiederhold-G.

Author affiliation

IBM European Networking Center, Heidelberg, Germany.

Source

Proceedings of the Fifteenth International Conference on Very Large **Data Bases,** Amsterdam, Netherlands, 22-25 Aug. 1989, p.337-46.
Published: Morgan Kaufmann, Palo Alto, CA, USA, 1989, xii+467 pp.

Publication year

1989.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

P Practical.

Abstract

A simple and efficient recovery method for **nested transactions,** called ARIES/NT (algorithm for recovery and isolation exploiting semantics for **nested transactions),** that uses write-ahead logging and supports semantically-rich modes of locking and operation logging is presented. This method applies to a very general model of **nested transactions,** which includes partial **rollbacks** of subtransactions, upward and downward inheritance of locks, and concurrent execution of ancestor and descendent subtransactions. The adopted system architecture also compasses aspects of distributed **data base** management. ARIES/NT is an extension of the ARIES recovery and concurrency control method developed recently for the single-level **transaction** model by Mohan, et al. in the IBM Research Report RJ6649. (20 refs).

Descriptors
concurrency-control; distributed-databases; transaction-processing.
Keywords
ARIES NT; recovery method; write ahead logging; **nested transactions;** semantically rich modes of locking; operation logging; partial **rollbacks;** inheritance; concurrent execution; distributed **data base** management; concurrency control method.
Classification codes
C6160B (Distributed DBMS).

☑ **document 3 of 3** Order Document
INSPEC - 1969 to date (INZZ)

Accession number & update
3067233, C88013852; 880000.
Title
Log-based recovery for **nested transactions.**
Author(s)
Moss-J-E-B; Ed. by Stocker-P-M; Kent-W; Hammersley-P.
Author affiliation
Dept of Comput & Inf Sci, Massachusetts Univ, Amherst, MA, USA.
Source
Proceedings of the Thirteenth International Conference on Very Large **Data Bases:** 1987 13th VLDB, Brighton, UK, 1-4 Sept. 1987, p.427-32.
Sponsors: Alvey, ICL, RTI, et al.
Published: Morgan Kaufmann, Los Altos, CA, USA, 1987, xii+518 pp
Translation of: C04.
Publication year
1987.
Language
EN.
Publication type
CPP Conference Paper.
Treatment codes
P Practical.
Abstract
Techniques similar to shadow pages have been suggested for use in **rollback** and crash recovery for **nested transactions.** However, undo /redo log methods have not been presented, though undo/redo logs are widely used for **transaction** recovery, and perhaps preferable to shadow methods. The author develops a scheme of **log-based** recovery for **nested transactions.** The resulting design is promising because it requires a relatively small number of extensions to a similar scheme of recovery for single-level **transactions.** (14 refs).
Descriptors
database-management-systems; system-recovery.
Keywords
nested transactions; shadow pages; **rollback;** crash recovery; undoredo logs; log **based** recovery.
Classification codes
C6150J (Operating systems).
C6160 **(Database** management systems (DBMS)).

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save locally as: PDF document search strategy: do not include the search strategy

order

Select All

USPTO Full Text Retrieval Options

document 1 of 9 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
7629745, B2003-06-6210L-189, C2003-06-5620-050; 20030526.

Title
Commitment of mobile distributed real-time **nested transaction.**

Author(s)
Liu-Yun-Sheng; Liao-Guo-Qiong; Li-Guo-Hui; Xia-Jia-Li.

Author affiliation
Sch of Comput Sci & Technol, Huazhong Univ of Sci & Technol, Wuhan, China.

Source
Journal-of-Software (China), vol.14, no.1, p.139-45, Jan. 2003. , Published: Science Press.

CODEN
RUXUEW.

ISSN
ISSN: 1000-9825.

Availability
SICI: 1000-9825(200301)14:1L.139:CMDR; 1-M.

Publication year
2003.

Language
CH.

Publication type
J Journal Paper.

Treatment codes
P Practical.

Abstract

For **transaction** mobility and the inherence limitations of wireless networks, traditional real-time **transaction** management mechanisms are incompetent to support the execution of mobile distributed real-time **transactions** in a mobile distributed computing environment. In this paper, the **commit** mechanism for mobile real-time **transactions** is studied. First, a **nested transaction** model **based** on functional alterative tasks is given by analyzing the characteristics of real-time **transactions** in a mobile distributed environment. Then a three-tier **commit** structure supporting the suggested model is presented. A three-phase real-time **commit** protocol 3PRTC (three-phase real-time **commit)** is also proposed, which can guarantee the atomicity and structural correctness of the mobile real-time **transactions.** By performance testing, it is shown that the suggested **transaction** model and its **commit** mechanism can improve the successful ratio of real-time **transactions.** (9 refs).

Descriptors

mobile-computing; protocols; real-time-systems; transaction-processing; wireless-LAN.

Keywords

mobile distributed real time **nested transaction; transaction** mobility; wireless networks; real time **transaction** management; **commit** mechanism; mobile computing; three tier **commit** structure; three phase real time **commit** protocol; 3PRTC; performance testing.

Classification codes

B6210L (Computer communications).
B6150M (Protocols).
C5620 (Computer networks and techniques).
C6150N (Distributed systems software).
C5640 (Protocols).

☑ **document 2 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

6572114, C2000-06-6160B-009; 20000401.

Title

Open **nested transaction:** a support for increasing performance and for multi-tier applications.

Author(s)

Saheb-M; Karoui-R; Sedillot-S; Ed. by Saake-G; Schwarz-K; Turker-C.

Author affiliation

Inst Nat de Recherche en Inf et Autom, Le Chesnay, France.

Source

Proceedings of TDD'99: **Transactions** and **Database** Dynamics - Eighth International Workshop on Foundations of Models and Languages for **Data** and Objects, Schloss Dagstuhl, Germany, 27-30 Sept. 1999.
In: p.115-38, 1999.

Publication year

1999.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

P Practical.

Abstract

The concept of a **transaction** has been developed to permit management of activities and resources in

a reliable computing environment. The two-phase **commit** protocol is combined with the strict two-phase locking protocol, as the means for ensuring atomicity and the serializability of **transactions.** The implication of this combination on the length of time a **transaction** may hold locks on various **data** items might be severe. There are certain classes of application where it is known that resources acquired within a **transaction** can be "released early", rather than having to wait until the **transaction** terminates. Furthermore, there are applications involving heterogeneous competing business organizations, which do not allow to block their resources; therefore, the preservation of local autonomy of individual systems is crucial. It is undesirable, for example, to use a protocol where a site belonging to a competing organization can harmfully block their local resources. Several enhancements to the traditional **transaction** model have been proposed either by relaxing the conventional ACID properties or by providing an asynchronous communication. This paper describes an extension of the OTS, by adding the Open **Nested Transaction** Model, which improves greatly **transaction** parallelism by releasing the **nested transaction** locks at the **nested transaction commit** time. Open **nested transactions** relax the isolation property by allowing the effects of the committed **nested transaction** to be visible to concurrent **transactions.** We describe also how we take the benefit of this model, through the proposed Asynchronous **Nested Transaction** model, to overcome the limits of the actual messaging products and standard specifications when they are confronted with the problem of guaranteeing the atomicity of distributed multi-tier **transactional** applications. (25 refs).

Descriptors

distributed-databases; memory-protocols; software-performance-evaluation; transaction-processing.

Keywords

performance; distributed multi tier **transactional** applications; two phase **commit** protocol; two phase locking protocol; atomicity; serializability; heterogeneous competing business organizations; protocol; Open **Nested Transaction** Model; **transaction** parallelism; **nested transaction** locks; Asynchronous **Nested Transaction** mode.

Classification codes

C6160B (Distributed **databases).**
C6150N (Distributed systems software).

☑ **document 3 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

5972390, C9808-6160J-015; 980715.

Title

A concurrency control algorithm for an open and safe **nested transaction** model.

Author(s)

Madria-S-K.

Author affiliation

Sch of Comput Sci, Univ Sains Malaysia, Penang, Malaysia.

Source

Proceedings of 1st International Conference on Information Communications and Signal Processing, vol.2, Singapore, 9-12 Sept. 1997.
In: p.907-12 vol.2, 1997.

ISSN

ISBN: 0-7803-3676-3, CCCC: 0 7803 3676 3/97/ ($10.00).

Publication year

1997.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

T Theoretical or Mathematical.

Abstract

We present a concurrency control algorithm for an open and safe **nested transaction** model. We use prewrite operations in our model to increase the concurrency. Prewrite operations are modeled as subtransactions in the **nested transaction** tree. The subtransaction which initiates prewrite subtransactions are modelled as recovery point subtransaction. The recovery point subtransaction can release their locks before its ancestors **commit.** Thus, our model increases the concurrency in comparison to other **nested transaction** models. Our model is useful an environment of long-running **transactions** common in object oriented **databases,** computer aided design and in the software development process. (29 refs).

Descriptors

CAD; concurrency-control; database-theory; object-oriented-databases; software-engineering.

Keywords

safe **nested transaction** model; open **nested transaction** model; concurrency control algorithm; prewrite operations; **nested transaction** tree; long running **transactions;** recovery point subtransaction; prewrite subtransactions; object oriented **databases;** computer aided design; software development process.

Classification codes

C6160J (Object-oriented **databases**).
C6110B (Software engineering techniques).
C4250 (**Database** theory).

☑ **document 4 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

5309668, C9608-6160-006; 960702.

Title

Commit scope control in **nested transactions.**

Author(s)

Qiming-Chen; Dayal-U; Ed. by Apers-S; Bouzeghoub-M; Gardarin-G.

Author affiliation

Hewlett-Packard Lab, Palo Alto, CA, USA.

Source

Proceedings of 5th Conference on Extended **Database** Technology (EDBT'96), Avignon, France, 25-29 March 1996.
In: p.621-4, 1996.

ISSN

ISBN: 3-540-61057-X.

Publication year

1996.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

P Practical; T Theoretical or Mathematical.

Abstract

A common limitation of all the existing **nested transaction** models is that they only allow

subtransactions to **commit** either to parent **transactions** or to **databases.** In order to adequately balance atomicity and concurrency at selected levels of a **transaction** hierarchy, the notion of scoped commitment is proposed, that allows a subtransaction to **commit** to a selected ancestor independently of its parent, making its results visible to that ancestor and thus improving the concurrency in the **transaction** subtree beneath that ancestor. A corresponding scoped undo approach is also developed that allows a **transaction** hierarchy with subtransactions having mixed **commit** scopes to partially and consistently roll back upon failure, then restart and roll forward. (4 refs).

Descriptors
concurrency-control; database-management-systems; database-theory; transaction-processing.

Keywords
commit scope control; **nested transactions;** subtransactions; parent **transactions; databases;** atomicity; concurrency control; **transaction** hierarchy; scoped commitment; selected ancestor; **transaction** subtree; scoped undo approach; mixed **commit** scopes; failure; restart; roll back; roll forward.

Classification codes

C6160 (Database management systems (DBMS)).
C4250 (Database theory).
C6130 (Data handling techniques).

USPTO Full Text Retrieval Options

☑ **document 5 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
5073017, C9511-6160B-023; 951004.

Title
Hybrid atomicity for **nested transactions.**

Author(s)
Fekete-A; Lynch-N; Weihl-W-E.

Author affiliation
Sydney Univ, NSW, Australia.

Source
ICDT 92. International Conference on **Database** Theory, Berlin, Germany, 14-16 Oct. 1992.
In: Theoretical-Computer-Science (Netherlands), vol.149, no.1, p.151-78, 18 Sept. 1995.

CODEN
TCSCDI.

ISSN
ISSN: 0304-3975, CCCC: 0304-3975/95/ ($09.50).

Publication year
1995.

Language
EN.

Publication type
CPP Conference Paper, J Journal Paper.

Treatment codes
P Practical.

Abstract
This paper defines the notion of hybrid atomicity for **nested transaction** systems, and presents and

verifies an algorithm providing this property. Hybrid atomicity is a modular property; it allows the correctness of a system to be deduced from the fact that each object is implemented to have the property. It allows more concurrency than dynamic atomicity, by assigning timestamps to **transactions** at **commit.** The Avalon system provides exactly this facility. The results in this paper extend earlier work using the same model for locking and **timestamp-based** algorithms, providing further evidence for the generality of the approach. However, there are some subtle differences with the definitions used in earlier work, showing the difficulties of developing precise general models for **nested transaction** systems. (16 refs).

Descriptors

abstract-data-types; concurrency-control; transaction-processing.

Keywords

hybrid atomicity; **nested transactions;** modular property; timestamps; Avalon system; locking; timestamp **based** algorithms; **nested transaction** systems.

Classification codes

C6160B (Distributed databases).
C6150J (Operating systems).
C6120 (File organisation).

---

document 6 of 9 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

4664013, C9406-6160K-009; 940427.

Title

A **semantic-based nested transaction** model for intelligent and cooperative information systems.

Author(s)

Haghjoo-M-S; Papazoglou-M-P; Schmidt-H-W; Ed. by Huhns-M; Papazoglou-M-P; Schlageter-G.

Author affiliation

Dept of Comput Sci, Australian Nat Univ, Canberra, ACT, Australia.

Source

Proceedings of International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Netherlands, 12-14 May 1993.
Sponsors: IEEE, ACM, AAAI, IFIP, IPSJ, NGI.
In: p.321-31, 1993.

ISSN

ISBN: 0-8186-3135-X, CCCC: 0 8186 3135 X/93/ ($3.00).

Publication year

1993.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

T Theoretical or Mathematical.

Abstract

Intelligent and cooperative information systems (ICIS) involve upgraded models for **transaction** support for large distributed **data** /knowledge intensive applications. The use of long-lived **transactions** results in the introduction of a flexible **transaction** model which employs weak integrity and optimistic **commit** mechanisms for **transactions** relying on compensating and contingency **transactions** to recover from potential semantic failures. An **open-nested transaction** model for

ICIS is introduced. It provides linguistic facilities for defining and controlling long-lived complex **transactions** (comprising related units of work) by taking into account the semantics of distributed computations. The fact that the proposed **transaction** model is grafted on to an appropriately extended concurrent object-oriented programming language allows exhibiting a high degree of parallelism inherent in distributed **transaction-oriented** applications. (28 refs).

Descriptors

cooperative-systems; database-theory; deductive-databases; distributed-databases; information-systems; object-oriented-languages; parallel-languages; system-recovery; transaction-processing.

Keywords

semantic **based nested transaction** model; intelligent information systems; cooperative information systems; upgraded models; **transaction** support; large distributed **data** applications; knowledge intensive applications; related work units; weak integrity; optimistic **commit** mechanisms; compensating **transactions;** contingency **transactions;** semantic failure recovery; open **nested transaction** model; linguistic facilities; long lived complex **transactions;** distributed computations; extended concurrent object oriented programming language; parallelism; distributed **transaction** oriented applications.

Classification codes

C6160K (Deductive databases).
C6110J (Object-oriented programming).
C6140D (High level languages).
C6160B (Distributed DBMS).
C4250 (Database theory).

USPTO Full Text Retrieval Options

☑ **document 7 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

4426791, C9307-6150N-038; 930610.

Title

Nested transaction based reliable distributed computing environment for a network of workstations.

Author(s)

Kanai-T; Shirakihara-T.

Author affiliation

Res & Dev Center, Toshiba Corp, Tokyo, Japan.

Source

Transactions-of-the-Information-Processing-Society-of-Japan (Japan), vol.33, no.11, p.1384-93, 1992.

CODEN

JSGRD5.

ISSN

ISSN: 0387-5806.

Publication year

1992.

Language

JA.

Publication type

J Journal Paper.

Treatment codes

P Practical.

Abstract

Describes an implementation method for distributed **transactions;** the functions needed for distributed

transaction processing; the system configuration; the structure of a multi-thread remote procedure call (RPC) server; message flow for distributed **transaction** processing; message flow for **commit** processing; an RPC specification; application programs for client and server; the performance of RPCs; the behaviour of the program to be measured; and the time required for **commit** processing. (14 refs).

Descriptors
distributed-processing; local-area-networks; remote-procedure-calls; transaction-processing; workstations.

Keywords
nested transactions; workstation network; multithread remote procedure call server; client server system; reliable distributed computing environment; distributed **transaction** processing; system configuration; message flow; **commit** processing; RPC specification; performance.

Classification codes
C6150N (Distributed systems).

---

document 8 of 9 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4310729, C9302-4250-021; 921216.

Title
Hybrid atomicity for **nested transactions.**

Author(s)
Fekete-A; Lynch-N; Weihl-W-E; Ed. by Biskup-J; Hull-R.

Author affiliation
Dept of Comput Sci, Sydney Univ, NSW, Australia.

Source
Database Theory - ICDT '92. 4th International Conference Proceedings, Berlin, Germany, 14-16 Oct. 1992, p.216-30.
Published: Springer-Verlag, Berlin, Germany, 1992, ix+449 pp.

ISSN
ISBN: 3-540-56039-4.

Publication year
1992.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
T Theoretical or Mathematical.

Abstract
Defines the notion of hybrid atomicity for **nested transaction** systems, and presents and verifies an algorithm providing this property. Hybrid atomicity is a modular property; it allows the correctness of a system to be deduced from the fact that each object is implemented to have the property. It allows more concurrency than dynamic atomicity, by assigning timestamps to **transactions** at **commit.** The Avalon system provides exactly this facility. (17 refs).

Descriptors
concurrency-control; database-theory; distributed-databases; program-verification; transaction-processing.

Keywords
algorithm verification; system correctness; **nested transactions;** hybrid atomicity; modular property; concurrency; timestamps; **commit;** Avalon system.

Classification codes

C4250 (Database theory).

C6160B (Distributed DBMS).
C4240 (Programming and algorithm theory).

USPTO Full Text Retrieval Options

☑ **document 9 of 9** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
3381584, C89038424; 890000.

Title
A concurrency control scheme for **nested transactions.**

Author(s)
Shin-D-C; Moon-S-C.

Author affiliation
Dept of Comput Sci, Korea Adv Inst of Sci & Technol, Seoul, South Korea.

Source
Fourteenth EUROMICRO Symposium on Microprocessing and Microprogramming (EUROMICRO '88), Zurich, Switzerland, 29 Aug.-1 Sept. 1988.
Sponsors: Bank Leu, Control **Data** (Schweiz), GEI Systeme, et al.
In: Microprocessing-Microprogramming (Netherlands), vol.25, no.1-5, p.233-8, Jan. 1989.

CODEN
MMICDT.

ISSN
ISSN: 0165-6074.

Publication year
1989.

Language
EN.

Publication type
CPP Conference Paper, J Journal Paper.

Treatment codes
P Practical.

Abstract
Locking rules for concurrency control in **nested transaction** model, in which parent/child parallelism, sibling parallelism, and arbitrary commitment-dependence relationship between **transactions** are allowed, are presented. The locking rules are **based** on the lock transformation scheme that never allows the **commit** deadlock to occur, which could happen by permitting parent/child parallelism. In addition, two proposed locking rules are compared to each other with respect to the overhead involved in cascading abort, the degree of concurrency, and the frequency of deadlock occurrences. (14 refs).

Descriptors
parallel-processing.

Keywords
locking rules; concurrency control scheme; **nested transactions;** parentchild parallelism; sibling parallelism; arbitrary commitment dependence relationship; cascading abort.

Classification codes
C5440 (Multiprocessor systems and techniques).

Save locally as: PDF document search strategy: do not include the search strategy

Top - News & FAQS - Dialog

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

save locally as: PDF document search strategy: do not include the search strategy

order

Select All

1 Commitment of mobile distributed real-time nested transaction.
2 Relaxed atomic commit for real-time transactions in mobile computing environ
3 A theory of transactions on recoverable search trees.
4 Research on the transaction commit mechanism for distributed enginee
5 Multi-agent cooperative transactions for e-commerce.
6 Open nested transaction: a support for increasing performance and for multi-t
7 A meta-object protocol for distributed OO applications.
8 Design and evaluation of a new transaction execution model for multidat
9 Commit scope control in nested transactions.
10 Hybrid atomicity for nested transactions.
11 Interoperability between PCTE and external database systems: standard transact
12 A semantic-based nested transaction model for intelligent and cooper
13 Transaction management in design databases.
14 Applications of transaction processing for session management in multi-me
15 Hybrid atomicity for nested transactions.
16 Cooperating transactions and workspaces in EPOS: design and prelimin
17 Unilateral commit: a new paradigm for reliable distributed transaction process
18 An approach to efficient distributed transactions.

USPTO Full Text Retrieval Options

document 1 of 18 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
7629745, B2003-06-6210L-189, C2003-06-5620-050; 20030526.

Title
Commitment of mobile distributed real-time **nested transaction.**

Author(s)
Liu-Yun-Sheng; Liao-Guo-Qiong; Li-Guo-Hui; Xia-Jia-Li.

Author affiliation
Sch of Comput Sci & Technol, Huazhong Univ of Sci & Technol, Wuhan, China.

Source
Journal-of-Software (China), vol.14, no.1, p.139-45, Jan. 2003. , Published: Science Press.

CODEN
RUXUEW.

ISSN
ISSN: 1000-9825.

Availability

SICI: 1000-9825(200301)14:1L.139:CMDR; 1-M.

Publication year
2003.

Language
CH.

Publication type
J Journal Paper.

Treatment codes
P Practical.

Abstract
For **transaction** mobility and the inherence limitations of wireless networks, traditional real-time **transaction** management mechanisms are incompetent to support the execution of mobile distributed real-time **transactions** in a mobile distributed computing environment. In this paper, the **commit** mechanism for mobile real-time **transactions** is studied. First, a **nested transaction** model **based** on functional alterative tasks is given by analyzing the characteristics of real-time **transactions** in a mobile distributed environment. Then a three-tier **commit** structure supporting the suggested model is presented. A three-phase real-time **commit** protocol 3PRTC (three-phase real-time **commit)** is also proposed, which can guarantee the atomicity and structural correctness of the mobile real-time **transactions.** By performance testing, it is shown that the suggested **transaction** model and its **commit** mechanism can improve the successful ratio of real-time **transactions.** (9 refs).

Descriptors
mobile-computing; protocols; real-time-systems; transaction-processing; wireless-LAN.

Keywords
mobile distributed real time **nested transaction; transaction** mobility; wireless networks; real time **transaction** management; **commit** mechanism; mobile computing; three tier **commit** structure; three phase real time **commit** protocol; 3PRTC; performance testing.

Classification codes

```
B6210L    (Computer communications).
B6150M    (Protocols).
C5620     (Computer networks and techniques).
C6150N    (Distributed systems software).
C5640     (Protocols).
```

☑ **document 2 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
7440411, C2002-12-6150N-064; 20021104.

Title
Relaxed atomic **commit** for real-time **transactions** in mobile computing environment.

Author(s)
YunSheng-Liu; GuoQiong-Liao; GuoHui-Li; JiaLi-Xia; Ed. by Meng-X; Su-J; Wang-Y.

Author affiliation
Coll of Comput Sci & Technol, Huazhong Univ of Sci & Technol, Hubei, China.

Source
Advances in Web-Age Information Management. Third International Conference, WAIM 2002. Proceedings, Beijing, China, 11-13 Aug. 2002.
Sponsors: Nat. Sci. Found. China, Microsoft Res. Asia, Oracle (China), IBM DB22(China).
In: p.397-408, 2002.

ISSN

ISBN: 3-540-44045-3.

Publication year
2002.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
P Practical.

Abstract
It is more difficult for real-time **transactions** in mobile computing environment (MCE) to meet their deadlines than that in traditional distributed computing environment due to the mobility of **transactions** and inherent limitations of wireless network. It is necessary to study new **transaction** management mechanisms for mobile real-time **transactions** (MRTTs) to make more of them successful. In the first of the paper, functional alternative tasks are suggested for important MRTTs in the interest of enhancing their reliability. Then a **nested** mobile real-time **transaction** model **based** on functional alternation is presented. For guaranteeing the relaxed atomicity and structure correctness of MRTTs, a Three-Phase Real-Time **Commit** protocol (3PRTC) taking the deadlines associated with MRTTs and mobility into consideration is proposed. In the end, the correctness of 3PRTC is proved. (14 refs).

Descriptors
mobile-computing; real-time-systems; transaction-processing; transport-protocols.

Keywords
real time **transactions;** mobile computing environment; wireless network; **transaction** management mechanisms; functional alternative tasks; **nested** mobile real time **transaction** model; functional alternation; relaxed atomicity; structure correctness; Three Phase Real Time **Commit** protocol; relaxed atomic **commit.**

Classification codes
C6150N (Distributed systems software).
C5640 (Protocols).

Copyright statement

☑ **document 3 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
7011434, C2001-09-4250-013; 20010813.

Title
A theory of **transactions** on recoverable search trees.

Author(s)
Sippu-S; Soisalon-Soininen-E; Ed. by Van-den-Bussche-J; Vianu-V.

Author affiliation
Dept of Comput Sci, Helsinki Univ, Finland.

Source
Proceedings of 8th International Conference on **Database** Theory, London, UK, 4-6 Jan. 2001.
Sponsors: Eur. Union, Eur. Res. Consortium for Inf. & Math.
In: p.83-98, 2001.

ISSN
ISBN: 3-540-41456-8.

Publication year
2001.

Language
EN.

Publication type
CPP Conference Paper.
Treatment codes
T Theoretical or Mathematical.
Abstract
We consider **transactions** running on a **database** that consists of records with unique totally-ordered keys and is organized as a sparse primary search tree such as a B-tree index on disk storage. We extend the classical read-write model of **transactions** by considering inserts, deletes and key-range scans and by distinguishing between four types of **transaction** states: forward-rolling, committed, backward-rolling, and rolled-back **transactions.** A search-tree **transaction** is modelled as a two-level **transaction** containing structure modifications as open **nested** subtransactions that can **commit** even though the parent **transaction** aborts. Isolation conditions are defined for search-tree **transactions** with **nested** structure modifications that guarantee the structural consistency of the search tree, a required isolation level (including phantom prevention) for **database** operations, and recoverability for structure modifications and **database** operations. (20 refs).
Descriptors
concurrency-control; database-theory; disc-storage; query-processing.
Keywords
theory of **transactions;** recoverable search trees; totally ordered keys; sparse primary search tree; B tree index; disk storage; classical read write model; inserts; deletes; key range scans; forward rolling; backward rolling; rolled back **transactions;** two level **transaction;** committed **transaction;** open **nested** subtransactions; **transaction** aborts; isolation conditions; structural consistency; phantom prevention; **database** operations; recoverability; structure modifications.
Classification codes
C4250 **(Database** theory).
C6160 **(Database** management systems (DBMS)).
Copyright statement

USPTO Full Text Retrieval Options

☑ **document 4 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
6934616, C2001-07-6160B-001; 20010528.
Title
Research on the **transaction commit** mechanism for distributed engineering **database** system.
Author(s)
Liao-Guo-Qiong; Li-Tao-Shen.
Author affiliation
Coll of Comput & Inf Eng, Guangxi Univ, Nanning, China.
Source
Journal-of-Computer-Aided-Design-Computer-Graphics (China), vol.13, no.4, p.357-61, April 2001. , Published: Science Press.
CODEN
JFTXFX.
ISSN
ISSN: 1003-9775.
Availability
SICI: 1003-9775(200104)13:4L.357:RTCM; 1-M.
Publication year
2001.
Language
CH.

Publication type
J Journal Paper.
Treatment codes
P Practical.
Abstract
Since the ACID (Atomic, Consistency, Isolation, Durability) properties of **transactions** in distributed engineering **databases** differ from that of conventional ones, the Basic Two-Phase **Commit** (B2PC) protocol no longer fits. **Based** on the **nested** model, we establish a structure for **transaction** management in a distributed engineering **database.** We also propose a new Two-Phase **Commit** protocol with a detailed description of its principle, algorithms and ways of handling failures. (4 refs).
Descriptors
distributed-databases; engineering-information-systems; memory-protocols; transaction-processing.
Keywords
transaction commit mechanism; distributed engineering **database;** ACID properties; **transaction** management; Basic Two Phase **Commit** protocol; **nested** model.
Classification codes
C6160B (Distributed **databases).**
C6120 (File organisation).
Copyright statement

☑ **document 5 of 18** Order Document
INSPEC - 1969 to date (INZZ)

Accession number & update
6911931, C2001-06-7180-010; 20010401.
Title
Multi-agent cooperative **transactions** for e-commerce.
Author(s)
Chen-Q; Dayal-U; Ed. by Etzion-O; Scheuermann-P.
Author affiliation
HP Labs, Hewlett-Packard Co, Palo Alto, CA, USA.
Source
Cooperative Information Systems. 7th International Conference, CoopIS 2000, Eilat, Israel, 6-8 Sept. 2000.
In: p.311-22, 2000.
ISSN
ISBN: 3-540-41021-X.
Publication year
2000.
Language
EN.
Publication type
CPP Conference Paper.
Treatment codes
T Theoretical or Mathematical.
Abstract
E-commerce is a distributed computing environment with dynamic relationships among a large number of autonomous service requesters, brokers and providers. With the increasing automation of e-commerce applications, we will see the use of software agents that cooperate to perform business **transactions.** Multi-agent cooperative **transactions** are different in their requirements both from conventional atomic **transactions** executed under centralized control and from **nested transactions** executed under hierarchical control. Cooperative **transactions** require peer-to-peer protocols for **commit** control and failure recovery. The significance and feasibility of this approach have been demonstrated in a prototype implemented at HP Labs, using our dynamic agent infrastructure. (12

refs).

Descriptors

electronic-commerce; multi-agent-systems; protocols; software-agents; system-recovery.

Keywords

multi agent cooperative **transactions;** e commerce; distributed computing environment; dynamic relationships; autonomous service requesters; autonomous service brokers; autonomous service providers; software agents; business **transactions;** atomic **transactions;** centralized control; **nested transactions;** hierarchical control; peer to peer protocols; **commit** control; failure recovery; dynamic agent infrastructure.

Classification codes

C7180 (Retailing and distribution computing).
C6170 (Expert systems and other AI software and techniques).
C5640 (Protocols).
C6155 (Computer communications software).
C6150G (Diagnostic, testing, debugging and evaluating systems).
C7120 (Financial computing).

---

document 6 of 18 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

6572114, C2000-06-6160B-009; 20000401.

Title

Open **nested transaction:** a support for increasing performance and for multi-tier applications.

Author(s)

Saheb-M; Karoui-R; Sedillot-S; Ed. by Saake-G; Schwarz-K; Turker-C.

Author affiliation

Inst Nat de Recherche en Inf et Autom, Le Chesnay, France.

Source

Proceedings of TDD'99: **Transactions** and **Database** Dynamics - Eighth International Workshop on Foundations of Models and Languages for **Data** and Objects, Schloss Dagstuhl, Germany, 27-30 Sept. 1999.
In: p.115-38, 1999.

Publication year

1999.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

P Practical.

Abstract

The concept of a **transaction** has been developed to permit management of activities and resources in a reliable computing environment. The two-phase **commit** protocol is combined with the strict two-phase locking protocol, as the means for ensuring atomicity and the serializability of **transactions.** The implication of this combination on the length of time a **transaction** may hold locks on various **data** items might be severe. There are certain classes of application where it is known that resources acquired within a **transaction** can be "released early", rather than having to wait until the **transaction** terminates. Furthermore, there are applications involving heterogeneous competing

business organizations, which do not allow to block their resources; therefore, the preservation of local autonomy of individual systems is crucial. It is undesirable, for example, to use a protocol where a site belonging to a competing organization can harmfully block their local resources. Several enhancements to the traditional **transaction** model have been proposed either by relaxing the conventional ACID properties or by providing an asynchronous communication. This paper describes an extension of the OTS, by adding the Open **Nested Transaction** Model, which improves greatly **transaction** parallelism by releasing the **nested transaction** locks at the **nested transaction commit** time. Open **nested transactions** relax the isolation property by allowing the effects of the committed **nested transaction** to be visible to concurrent **transactions.** We describe also how we take the benefit of this model, through the proposed Asynchronous **Nested Transaction** model, to overcome the limits of the actual messaging products and standard specifications when they are confronted with the problem of guaranteeing the atomicity of distributed multi-tier **transactional** applications. (25 refs).

Descriptors

distributed-databases; memory-protocols; software-performance-evaluation; transaction-processing.

Keywords

performance; distributed multi tier **transactional** applications; two phase **commit** protocol; two phase locking protocol; atomicity; serializability; heterogeneous competing business organizations; protocol; Open **Nested Transaction** Model; **transaction** parallelism; **nested transaction** locks; Asynchronous **Nested Transaction** mode.

Classification codes

C6160B (Distributed **databases).**
C6150N (Distributed systems software).

Copyright statement

☑ **document 7 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

5869825, C9805-6150N-010; 980324.

Title

A meta-object protocol for distributed OO applications.

Author(s)

Scinturier-L; Duchien-L; Florin-G; Ed. by Ege-R; Singh-M; Meyer-B.

Author affiliation

CNAM-Lab CEDRIC, Paris, France.

Source

Proceedings of TOOLS USA 97. International Conference on Technology of Object Oriented Systems and Languages, Santa Barbara, CA, USA, 28 July-1 Aug. 1997.
Sponsors: Interactive Software Eng.
In: p.306-17, 1998.

ISSN

ISBN: 0-8186-8383-X, CCCC: 0 8186 8383 X/98/ ($10.00).

Publication year

1998.

Language

EN.

Publication type

CPP Conference Paper.

Treatment codes

P Practical.

Abstract

The design of complex distributed object applications such as **transactional** systems (e.g. **nested transactions,** two or three phase **commit** protocols), network algorithms (e.g. routing, spanning tree construction, group causal delivery) or cooperative applications is a hard task. We present a meta-

object protocol (MOP) called CAO-LAC, to assist developers in implementing these applications. A prototype was developed for the language of the GUIDE distributed object system. The MOP uses a state/transition approach to synchronize concurrent objects. We propose some extensions to manage intra-object parallelism. Then we report on a spanning tree construction algorithm that was implemented with this MOP. (27 refs).

Descriptors

distributed-processing; object-oriented-languages; object-oriented-programming; protocols; synchronisation; trees-mathematics.

Keywords

meta object protocol; distributed object oriented applications; **transactional** systems; **nested transactions;** three phase **commit** protocols; two phase **commit** protocols; network algorithms; routing; spanning tree; group causal delivery; cooperative applications; CAO LAC; prototype; GUIDE; state transition approach; concurrent object synchronization; intra object parallelism.

Classification codes

C6150N (Distributed systems software).
C6110J (Object-oriented programming).

Copyright statement

Copyright 1998, IEE.

☑ **document 8 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update

5706260, C9711-6160B-009; 970930.

Title

Design and evaluation of a new **transaction** execution model for multidatabase systems.

Author(s)

Devirmis-T; Ulusoy-O.

Author affiliation

Dept of Comput Eng & Inf Sci, Bilkent Univ, Ankara, Turkey.

Source

Information-Sciences (USA), vol.102, no.1-4, p.203-38, Nov. 1997. , Published: Elsevier.

CODEN

ISIJBC.

ISSN

ISSN: 0020-0255, CCCC: 0020-0255/97/ ($17.00).

Availability

SICI: 0020-0255(199711)102:1/4L.203:DETE; 1-V
Electronic Journal Document Number: S0020-0255(97)00015-7.

Publication year

1997.

Language

EN.

Publication type

J Journal Paper.

Treatment codes

T Theoretical or Mathematical.

Abstract

In this paper, we present a new **transaction** execution model that captures the formalism and semantics of various extended **transaction** models and adopts them to a multidatabase system (MDBS) environment. The proposed model covers **nested transactions,** various dependency types among **transactions,** and **commit** independent **transactions.** The formulation of complex MDBS **transaction** types can be accomplished easily with the extended semantics captured in the model. A

detailed performance model of an MDBS is employed in investigating the performance implications of the proposed **transaction** model. (19 refs).

Descriptors
database-theory; distributed-databases; transaction-processing.

Keywords
transaction execution model; multidatabase systems; formalism; semantics; extended **transaction** models; multidatabase system; MDBS; **nested transactions;** dependency types; **commit** independent **transactions;** performance model.

Classification codes
C6160B (Distributed **databases).**
C4250 **(Database** theory).

Copyright statement

☑ **document 9 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
5309668, C9608-6160-006; 960702.

Title
Commit scope control in **nested transactions.**

Author(s)
Qiming-Chen; Dayal-U; Ed. by Apers-S; Bouzeghoub-M; Gardarin-G.

Author affiliation
Hewlett-Packard Lab, Palo Alto, CA, USA.

Source
Proceedings of 5th Conference on Extended **Database** Technology (EDBT'96), Avignon, France, 25-29 March 1996.
In: p.621-4, 1996.

ISSN
ISBN: 3-540-61057-X.

Publication year
1996.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
P Practical; T Theoretical or Mathematical.

Abstract
A common limitation of all the existing **nested transaction** models is that they only allow subtransactions to **commit** either to parent **transactions** or to **databases.** In order to adequately balance atomicity and concurrency at selected levels of a **transaction** hierarchy, the notion of scoped commitment is proposed, that allows a subtransaction to **commit** to a selected ancestor independently of its parent, making its results visible to that ancestor and thus improving the concurrency in the **transaction** subtree beneath that ancestor. A corresponding scoped undo approach is also developed that allows a **transaction** hierarchy with subtransactions having mixed **commit** scopes to partially and consistently roll back upon failure, then restart and roll forward. (4 refs).

Descriptors
concurrency-control; database-management-systems; database-theory; transaction-processing.

Keywords
commit scope control; **nested transactions;** subtransactions; parent **transactions; databases;** atomicity; concurrency control; **transaction** hierarchy; scoped commitment; selected ancestor; **transaction** subtree; scoped undo approach; mixed **commit** scopes; failure; restart; roll back; roll

forward.

Classification codes

C6160 (**Database** management systems (DBMS)).
C4250 (**Database** theory).
C6130 (**Data** handling techniques).

---

document 10 of 18 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
5073017, C9511-6160B-023; 951004.

Title
Hybrid atomicity for **nested transactions.**

Author(s)
Fekete-A; Lynch-N; Weihl-W-E.

Author affiliation
Sydney Univ, NSW, Australia.

Source
ICDT 92. International Conference on **Database** Theory, Berlin, Germany, 14-16 Oct. 1992.
In: Theoretical-Computer-Science (Netherlands), vol.149, no.1, p.151-78, 18 Sept. 1995.

CODEN
TCSCDI.

ISSN
ISSN: 0304-3975, CCCC: 0304-3975/95/ ($09.50).

Publication year
1995.

Language
EN.

Publication type
CPP Conference Paper, J Journal Paper.

Treatment codes
P Practical.

Abstract
This paper defines the notion of hybrid atomicity for **nested transaction** systems, and presents and verifies an algorithm providing this property. Hybrid atomicity is a modular property; it allows the correctness of a system to be deduced from the fact that each object is implemented to have the property. It allows more concurrency than dynamic atomicity, by assigning timestamps to **transactions** at **commit.** The Avalon system provides exactly this facility. The results in this paper extend earlier work using the same model for locking and **timestamp-based** algorithms, providing further evidence for the generality of the approach. However, there are some subtle differences with the definitions used in earlier work, showing the difficulties of developing precise general models for **nested transaction** systems. (16 refs).

Descriptors
abstract-data-types; concurrency-control; transaction-processing.

Keywords
hybrid atomicity; **nested transactions;** modular property; timestamps; Avalon system; locking; timestamp **based** algorithms; **nested transaction** systems.

Classification codes

C6160B (Distributed databases).
C6150J (Operating systems).
C6120 (File organisation).

Copyright statement
Copyright 1995, IEE.

---

document 11 of 18 Order Document
INSPEC - 1969 to date (INZZ)

Accession number & update
4941359, C9506-6160B-005; 950511.
Title
Interoperability between PCTE and external **database** systems: standard **transactions** and beyond.
Author(s)
Gabriel-P; Micknis-S; Schween-H.
Author affiliation
ISST, Fraunhofer Inst for Software Eng & Syst Eng, Berlin, Germany.
Source
Proceedings of PCTE '94, San Francisco, CA, USA, 29 Nov.-1 Dec. 1994.
In: PCTE-Technical-Journal (UK), no.2, p.351-68, 1994.
Publication year
1994.
Language
EN.
Publication type
CPP Conference Paper, J Journal Paper.
Treatment codes
P Practical.
Abstract
In order to cope with consistency-preserving operations (i.e. **transactions)** over distributed and heterogeneous **database** systems, all **database** systems involved must support a certain **transaction** protocol. Unfortunately, the ECMA-PCTE standard does not contain a protocol of this kind. The common protocol for distributed **transactions** is the two-phase **commit.** An increasingly accepted industrial standard for distributed **transactions** covering the two-phase **commit** is the XA specification by the X/Open group which is supported by most of the **UNIX-database** vendors. In this paper, we propose a concept for an interface that fulfils the XA specification and can be implemented on top of a system conforming to the ECMA-PCTE standard. Recently, more and more (distributed) applications have been **based** on **(transactions** on) **database** systems, e.g. CAD/CAM systems or software engineering environments which are typical PCTE applications. The two-phase **commit** protocol is not always adequate for all these often long-lived kinds of **transactions.** A great number of non-standard **transactions** have been suggested to overcome problems like inefficient blocking of **data** or lack of cooperation between complex activities. However, currently available **transaction** managers just support the two-phase **commit.** In the second part of the paper, we present a concept for a **transaction** system which allows the processing of both standard and non-standard **transactions** (including **nested transactions,** SAGAS, split-and-join **transactions** and **S-transactions)** over different and distributed **database** systems. (16 refs).
Descriptors
distributed-databases; open-systems; programming-environments; protocols; software-portability; transaction-processing.
Keywords
interoperability; two phase **commit** protocol; external **database** systems; standard **transactions;** consistency preserving operations; distributed **database** systems; heterogeneous **database** systems; **transaction** protocol; ECMA PCTE standard; XA specification; CAD CAM systems; software engineering environments; inefficient **data** blocking; cooperation; **nested transactions;** SAGAS; split and join

transactions; S **transactions;** CORBA.

Classification codes

C6160B (Distributed databases).
C6115 (Programming support).
C5640 (Protocols).

☑ **document 12 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4664013, C9406-6160K-009; 940427.

Title
A **semantic-based nested transaction** model for intelligent and cooperative information systems.

Author(s)
Haghjoo-M-S; Papazoglou-M-P; Schmidt-H-W; Ed. by Huhns-M; Papazoglou-M-P; Schlageter-G.

Author affiliation
Dept of Comput Sci, Australian Nat Univ, Canberra, ACT, Australia.

Source
Proceedings of International Conference on Intelligent and Cooperative Information Systems, Rotterdam, Netherlands, 12-14 May 1993.
Sponsors: IEEE, ACM, AAAI, IFIP, IPSJ, NGI.
In: p.321-31, 1993.

ISSN
ISBN: 0-8186-3135-X, CCCC: 0 8186 3135 X/93/ ($3.00).

Publication year
1993.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
T Theoretical or Mathematical.

Abstract
Intelligent and cooperative information systems (ICIS) involve upgraded models for **transaction** support for large distributed **data** /knowledge intensive applications. The use of long-lived **transactions** results in the introduction of a flexible **transaction** model which employs weak integrity and optimistic **commit** mechanisms for **transactions** relying on compensating and contingency **transactions** to recover from potential semantic failures. An **open-nested transaction** model for ICIS is introduced. It provides linguistic facilities for defining and controlling long-lived complex **transactions** (comprising related units of work) by taking into account the semantics of distributed computations. The fact that the proposed **transaction** model is grafted on to an appropriately extended concurrent object-oriented programming language allows exhibiting a high degree of parallelism inherent in distributed **transaction-oriented** applications. (28 refs).

Descriptors
cooperative-systems; database-theory; deductive-databases; distributed-databases; information-systems; object-oriented-languages; parallel-languages; system-recovery; transaction-processing.

Keywords
semantic **based nested transaction** model; intelligent information systems; cooperative information systems; upgraded models; **transaction** support; large distributed **data** applications; knowledge

intensive applications; related work units; weak integrity; optimistic **commit** mechanisms; compensating **transactions;** contingency **transactions;** semantic failure recovery; open **nested transaction** model; linguistic facilities; long lived complex **transactions;** distributed computations; extended concurrent object oriented programming language; parallelism; distributed **transaction** oriented applications.

Classification codes

C6160K (Deductive **databases**).
C6110J (Object-oriented programming).
C6140D (High level languages).
C6160B (Distributed DBMS).
C4250 (**Database** theory).

☑ **document 13 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4476179, C9310-6160Z-008; 930902.

Title
Transaction management in design **databases.**

Author(s)
Kumar-M; Wong-J.

Author affiliation
Dept of Comput Sci, Iowa State Univ, Ames, IA, USA.

Source
Journal-of-Systems-and-Software (USA), vol.22, no.1, p.3-15, July 1993.

CODEN
JSSODM.

ISSN
ISSN: 0164-1212, CCCC: 0164-1212/93/ ($6.00).

Publication year
1993.

Language
EN.

Publication type
J Journal Paper.

Treatment codes
A Application; P Practical.

Abstract
Conventional **data base** systems are not suitable for handling advanced applications encountered in engineering, such as CAD/CAM, CASE, CAE, and VLSI design. The **data bases** in such environments, also called design **data bases,** are characterized by the presence of many complex **data** objects denoted by a large number of small tables, as opposed to a few large tables encountered in conventional **data bases.** The **transaction** model used in tracking **data bases** for banking, inventory control, and other such applications use view serializability as the correctness criterion. While view serializability is appropriate for tracking **data bases,** it is unnecessarily restrictive for engineering **data bases.** The authors propose a **transaction** model that is suitable for accessing shared design **data based** on coserializability. The **transaction** model supports long-duration **transactions** with intermediate **commit** points. In addition to the conventional **nested transaction** hierarchy, the model allows for cooperation between **nested** design **transactions** needed in a design environment. (20 refs).

Descriptors
CAD-CAM; database-management-systems; transaction-processing.

Keywords
design **databases;** CAD CAM; CASE; CAE; VLSI design; **transaction** model; banking; inventory control; shared design **data;** coserializability; long duration **transactions.**
Classification codes

C6160Z (Other DBMS).
C6150J (Operating systems).
C7480 (Production engineering).

☑ **document 14 of 18** Order Document
INSPEC - 1969 to date (INZZ)

Accession number & update
4468054, B9310-6210L-050, C9310-5620-033; 930811.
Title
Applications of **transaction** processing for session management in multi-media information networks.
Author(s)
Kramer-M; Mohan-S.
Author affiliation
Bellcore, Morristown, NJ, USA.
Source
GLOBECOM '92. Communication for Global Users. IEEE Global Telecommunications Conference. Conference Record. (Cat. No.92CH3130-2) , Orlando, FL, USA, 6-9 Dec. 1992, p.764-9 vol.2.
Sponsors: IEEE.
Published: IEEE, New York, NY, USA, 1992, 3 vol. xlviii+1920 pp.
ISSN
ISBN: 0-7803-0608-2, CCCC: 0 7803 0608 2/92/ ($3.00).
Publication year
1992.
Language
EN.
Publication type
CPP Conference Paper.
Treatment codes
P Practical.
Abstract
Session management of complex multimedia calls a network-provided **transaction-oriented** service to a client is discussed. The authors suggest a set of goal-oriented policies that modify the all-or-nothing **transaction** semantics inherent in traditional **transaction** processing to accommodate more optimistic **commit** protocols that take corrective actions later if resources are found to be in inconsistent states. The use of **semantics-based** and **nested transactions** in allocating network resources is also suggested. These modifications are expected to improve overall resource utilization and reduce latency. (12 refs).
Descriptors
multimedia-systems; protocols; telecommunication-network-management; transaction-processing.
Keywords
network resource allocation; **transaction** processing; session management; information networks; complex multimedia; goal oriented policies; **commit** protocols.
Classification codes

B6210L (Computer communications).
B6210C (Network management).
B6150M (Protocols).

C5620 (Computer networks and techniques).
C5640 (Protocols).
C6150N (Distributed systems).

---

document 15 of 18 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4310729, C9302-4250-021; 921216.

Title
Hybrid atomicity for **nested transactions.**

Author(s)
Fekete-A; Lynch-N; Weihl-W-E; Ed. by Biskup-J; Hull-R.

Author affiliation
Dept of Comput Sci, Sydney Univ, NSW, Australia.

Source
Database Theory - ICDT '92. 4th International Conference Proceedings, Berlin, Germany, 14-16 Oct. 1992, p.216-30.
Published: Springer-Verlag, Berlin, Germany, 1992, ix+449 pp.

ISSN
ISBN: 3-540-56039-4.

Publication year
1992.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
T Theoretical or Mathematical.

Abstract
Defines the notion of hybrid atomicity for **nested transaction** systems, and presents and verifies an algorithm providing this property. Hybrid atomicity is a modular property; it allows the correctness of a system to be deduced from the fact that each object is implemented to have the property. It allows more concurrency than dynamic atomicity, by assigning timestamps to **transactions** at **commit.** The Avalon system provides exactly this facility. (17 refs).

Descriptors
concurrency-control; database-theory; distributed-databases; program-verification; transaction-processing.

Keywords
algorithm verification; system correctness; **nested transactions;** hybrid atomicity; modular property; concurrency; timestamps; **commit;** Avalon system.

Classification codes

C4250 (Database theory).
C6160B (Distributed DBMS).
C4240 (Programming and algorithm theory).

---

document 16 of 18 Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4117507, C9205-6110B-034; 920320.
Title
Cooperating **transactions** and workspaces in EPOS: design and preliminary implementation.
Author(s)
Conradi-R; Malm-C-C; Ed. by Andersen-R; Bubenko-J-A-Jr; Solvberg-A.
Author affiliation
Norwegian Inst of Technol, Trondheim, Norway.
Source
Advanced Information Systems Engineering. Third International Conference CAiSE '91 Proceedings, Trondheim, Norway, 13-15 May 1991, p.375-92.
Sponsors: Andersen Consulting.
Published: Springer-Verlag, Berlin, Germany, 1991, vi+578 pp.
ISSN
ISBN: 3-540-54059-8.
Publication year
1991.
Language
EN.
Publication type
CPP Conference Paper.
Treatment codes
P Practical.
Abstract
EPOS offers change-oriented versioning (COV) for software configuration management (CM). The EPOSDB has long and **nested transactions.** EPOS also supports software process management (PM) within a **transaction** and its workspace through task networks and their project infrastructure. The paper deals with EPOS extensions for **inter-transaction** coordination. This relies on intentional configuration descriptions and ambitions to describe change propagation into other versions. Raw (textual) merging comes for free in the COV model. Semantic merging is facilitated by **pre-commit** propagation and negotiation among overlapping **transactions,** according to agreed-upon protocols. (21 refs).
Descriptors
configuration-management; expert-systems; software-engineering; systems-analysis; transaction-processing.
Keywords
expert system for program and system development; semantic merging; EPOS; change oriented versioning; software configuration management; **nested transactions;** software process management; task networks; project infrastructure; inter **transaction** coordination; intentional configuration descriptions; ambitions; protocols.
Classification codes

```
C6110B    (Software engineering techniques).
C6170     (Expert systems).
C0310F    (Software development management).
```

☑ **document 17 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
4012715, C91072726; 911114.
Title

Unilateral **commit:** a new paradigm for reliable distributed **transaction** processing.

Author(s)
Hsu-M; Silberschatz-A.

Author affiliation
Digital Equipment Corp, Mountain View, CA, USA.

Source
Proceedings. Seventh International Conference on **Data** Engineering (Cat. No.91CH2968-6), Kobe, Japan, 8-12 April 1991, p.286-93.
Sponsors: IEEE.
Published: IEEE Comput. Soc. Press, Los Alamitos, CA, USA, 1991, xviii +766 pp.

ISSN
ISBN: 0-8186-2138-9, CCCC: CH2968-6/91/0000-0286 ($01.00).

Publication year
1991.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
P Practical.

Abstract
An alternative approach to distributed **transaction** processing **based** on the unilateral **commit** paradigm (UCP) and on persistent transmission is proposed. Instead of executing a unit of work as a single distributed **transaction,** as in the traditional **transaction** execution paradigm, opportunities are looked for to execute it as a structured set or a sequence of smaller, possibly single-site atomic **transactions.** Each such **transaction,** once executed, is committed independently of other **transactions** in the task. A method for rigorously maintaining the linkage between the steps is provided for by a persistent transmission mechanism. It is argued that UCP is especially attractive since it relies on a site's ability to execute conventional flat local **transactions** and does not require additional capabilities such as the ability to execute **nested transactions.** (23 refs).

Descriptors
distributed-databases; transaction-processing.

Keywords
paradigm; reliable distributed **transaction** processing; unilateral **commit** paradigm; structured set; atomic **transactions.**

Classification codes
C6160B (Distributed DBMS).

☑ **document 18 of 18** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
3263252, C89003421; 890000.

Title
An approach to efficient distributed **transactions.**

Author(s)
Ancilotti-P; Bertolino-A; Fusani-M.

Author affiliation
Pisa Univ, Italy.

Source
Distributed-Computing (West Germany), vol.2, no.4, p.201-12, 1988.

CODEN
DICOEB.

ISSN
ISSN: 0178-2770.
Publication year
1988.
Language
EN.
Publication type
J Journal Paper.
Treatment codes
P Practical.
Abstract
Most distributed systems proposed on the basis of the concept of atomic action or **transaction** strongly limit parallelism, thus reducing their level of efficiency. Features of efficiency in a distributed **transaction** system are investigated. Two mechanisms are proposed in order to enhance potential concurrency both among different **transactions** and within a single **transaction** during the **commit** phase: a synchronization mechanism has been designed which suggests an approach to concurrency control by allowing the release of acquired locks before **transaction** completion. The possibility of exploiting this mechanism to implement **nested transactions** is also discussed; and a distributed **commit** protocol is developed which enhances concurrency among the participants in an atomic action, thus achieving quick execution with high modularity. (19 refs).
Descriptors
distributed-processing; operating-systems-computers; parallel-programming; protocols; synchronisation.
Keywords
efficient distributed **transactions;** atomic action; **transaction;** parallelism; distributed **transaction** system; concurrency; synchronization mechanism; locks; **nested transactions;** distributed **commit** protocol.
Classification codes
C6150J (Operating systems).

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

☑ **document 1 of 1** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
7535742, C2003-03-4250-008; 20030224.

Title
Simulation of advanced **transaction** models using GOLOG.

Author(s)
Kiringa-I; Ed. by Ghelli-G; Grahne-G.

Author affiliation
Dept of Comput Sci, Toronto Univ, Ont, Canada.

Source
Database Programming Languages. 8th International Workshop, DBPL 2001. Revised Papers, Frascati, Italy, 8-10 Sept. 2001.
In: p.318-41, 2002.

ISSN
ISBN: 3-540-44080-1.

Publication year
2002.

Language
EN.

Publication type
CPP Conference Paper.

Treatment codes
T Theoretical or Mathematical.

Abstract
We propose a logical framework for describing, reasoning about, and simulating **transaction** models that relax some of the ACID (atomicity-consistency-isolation-durability) properties of classical **transactions.** Such extensions, usually called advanced **transaction** models (ATMs), have been proposed for dealing with new **database** applications involving long-lived, endless, and cooperative activities. Our approach appeals to non-Markovian theories, in which one may refer to past states other than the previous one. We specify an ATM as a suitable non-Markovian theory of the situation calculus, and its properties, including the relaxed ACID properties, as formulas of the same calculus. We use our framework to formalize classical and closed **nested transactions.** We first formulate each ATM and its properties as a theory of a certain kind and formulas of the situation calculus, respectively. We then define a legal **database** log as one whose actions are all possible and in which all the **commit** and **rollback** actions must occur whenever they are possible. After that, we show that the known properties of the ATM, including the (possibly relaxed) ACID constraints, are properties of legal logs and logical consequences of the theory corresponding to that ATM. Finally, we show how to use such a

specification as a **background** theory for **transaction** programs written in the situation calculus **based** programming language GOLOG. (23 refs).

Descriptors

concurrency-control; database-languages; database-theory; digital-simulation; relational-algebra; transaction-processing.

Keywords

advanced **transaction** models; GOLOG; simulation; reasoning; relaxed ACID properties; atomicity consistency isolation durability properties; **database** applications; nonMarkovian theories; past states; closed **nested transactions;** classical **transactions;** legal **database** log; **rollback** actions; **commit** actions; **transaction** programs; situation calculus **based** programming language.

Classification codes

C4250 (**Database** theory).
C6140D (High level languages).
C6160 (**Database** management systems (DBMS)).
C4210 (Formal logic).

Document

Select the documents you wish to save or order by clicking the box next to the document, or click the link above the document to order directly.

☑ **document 7 of 12** Order Document

INSPEC - 1969 to date (INZZ)

Accession number & update
6619905, B2000-07-6150M-074, C2000-07-5640-068; 20000601.

Title
Hybrid concurrency control and recovery for **multi-level** transactions.

Author(s)
Schewe-K-D; Ripke-T; Drechsler-S.

Author affiliation
Inst of Comput Sci, Tech Univ Clausthal, Germany.

Source
Acta-Cybernetica (Hungary), vol.14, no.3, p.419-53, 2000. , Published: Jozsef Attila Univ. Dept. Inf.

CODEN
ACCYDX.

ISSN
ISSN: 0324-721X.

Availability
SICI: 0324-721X(2000)14:3L.419:HCCR; 1-P.

Publication year
2000.

Language
EN.

Publication type
J Journal Paper.

Treatment codes
P Practical.

Abstract
Multi-level transaction schedulers adapt conflict-serializability on different **levels.** They exploit the fact that many **low-level** conflicts (e.g. on the **level** of pages) become irrelevant, if **higher-level** application semantics is taken into account. **Multi-level** transactions may lead to an increase in concurrency. It is easy to generalize locking protocols to the case of **multi-level** transactions. In this, however, the possibility of deadlocks may diminish the increase in concurrency. This stimulates the investigation of optimistic or hybrid approaches to concurrency control. Until now no hybrid concurrency control protocol for **multi-level** transactions has been published. The new FoPL protocol (Forward oriented Concurrency Control with Preordered Locking) is such a protocol. It employs access lists on the database objects and forward oriented **commit** validation. The basic test on all **levels** is

based on the reordering of the access lists. When combined with queueing and deadlock detection, the protocol is not only sound, but also complete for **multi-level** serializable schedules. This is definitely an advantage of FoPL compared with locking protocols. The complexity of deadlock detection is not crucial, since waiting transactions do not hold locks on database objects. Furthermore, the basic FoPL protocol can be optimized in various ways. Since the concurrency control protocol may force transactions to be aborted, it is necessary to support operation logging, It is shown that as well as **multi-level** locking protocols can be easily coupled with the ARIES algorithms. This also solves the problem of **rollback** during normal processing and crash recovery. (24 refs).

Descriptors

computational-complexity; concurrency-control; protocols; system-recovery.

Keywords

hybrid concurrency control; recovery; multi **level** transactions; transaction schedulers; conflict serializability; low **level** conflicts; application semantics; locking protocols; FoPL protocol; database objects; forward oriented **commit** validation; queueing; deadlock detection; complexity; ARIES algorithms.

Classification codes

```
B6150M    (Protocols).
C5640     (Protocols).
C6150J    (Operating systems).
C6150N    (Distributed systems software).
C4240C    (Computational complexity).
C6150G    (Diagnostic, testing, debugging and evaluating systems).
```

WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Thursday, August 18, 2005

Hide?	Set Name	Query	Hit Count
		DB=USPT; PLUR=NO; OP=OR	
☐	L84	L83 and (tree$ or hierarch$ or layer$ or level$ or node$ or branch$)	12
☐	L83	L82 not (ibm or (international adj1 business adj1 machine$))	12
☐	L82	L81 and (commit$ near transaction$)	27
☐	L81	L80 and ((start$ or abort$ or complet$ or end$ or rollback$ or (roll$ adj1 back$) or roll-back$ or roll$-back$) near transaction$)	38
☐	L80	(l67 or l77 or l78 or L79) and (nest$ near (database$ or (data adj1 base$) or transaction$))	52
☐	L79	707/201-206.ccls.	3501
☐	L78	707/103r-104.1.ccls.	3420
☐	L77	707/100.ccls.	1767
☐	L76	707/1.ccls.	1801
☐	L75	(L73 or L74) and (commit or committransaction or (commit adj1 transaction) or commit-transaction)	4
☐	L74	(L36 or L37) and (counter or counters).ab.	12
☐	L73	(L36 or L37) and (counter or counters).ti.	4
☐	L72	L71 and (commit or committransaction or (commit adj1 transaction) or commit-transaction)	19
☐	L71	(L69 or L70) and (rollback$ or (roll$ adj1 back$) or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	20
☐	L70	L68 and transaction$.ab.	53
☐	L69	L68 and transaction$.ti.	16
☐	L68	L67 not (ibm or (international adj1 business adj1 corporation))	155
☐	L67	(L36 or L37) and counter$	285
☐	L66	L65 not (ibm or (international adj1 business adj1 corporation))	5
☐	L65	L54 and counter$	19
☐	L64	L56 and counter$	5
☐	L63	(L44 or L45 or L58 or L62) and counter$	0
☐	L62	6922792.pn.	1
☐	L61	L58 and (roll$ near back$)	1
☐	L60	L58 and (rollback$ or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	0
☐	L59	L58 and (rollback or rollbacktransaction or (rollback adj1 transaction) or rollback-	0

10/055,208

		transaction)	
☐	L58	5878206.pn.	1
☐	L57	L56 and (rollback or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	14
☐	L56	L55 not (ibm or (international adj1 business adj1 corporation))	24
☐	L55	L54 and (hierarch$ or level$ or branch$ or node$ or layer$)	63
☐	L54	(L36 or L37) and (nest$ near transaction$)	65
☐	L53	L52 and end	8
☐	L52	L40 and start	9
☐	L51	L49 and (nest$ or hierarch$ or layer$ or level$ or node$ or branch$).ab.	3
☐	L50	L49 and (nest$ or hierarch$ or layer$ or level$ or node$ or branch$).ti.	0
☐	L49	L48 and (end near transaction)	30
☐	L48	(L36 or L37) and (start near transaction)	79
☐	L47	(L44 or L45) and (rollback or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	0
☐	L46	(L44 or L45) and (commit or committransaction or (commit adj1 transaction) or commit-transaction)	2
☐	L45	6751617.pn.	1
☐	L44	5778179.pn.	1
☐	L43	L42 and (commit or committransaction or (commit adj1 transaction) or commit-transaction)	5
☐	L42	L40 and (rollback or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	5
☐	L41	(L36 or L37) and ((nest$ near (window or windows or menu$ or icon$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or internet or (world adj1 wide adj1 web) or www or hypertext or hyperlink$ or http or url$)) not (ibm or (international adj1 business adj1 corporation)))	0
☐	L40	(L36 or L37) and nest$.ab.	15
☐	L39	(L36 or L37) and nest$.ti.	5
☐	L38	(L36 or L37) and ((navigat$ near (window or windows or menu$ or icon$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or internet or (world adj1 wide adj1 web) or www or hypertext or hyperlink$ or http or url$)) not (ibm or (international adj1 business adj1 corporation)))	11

(L36).pn. (6016394 5958010 5375216 5261089 5701480 5999931 6058389 5495606 5261069 5729738 5903891 6038562 6671686 5561763 5835908 4881166 5293627 5625815 5740434 5778388 5870763 5933838 5946698 6067550 6101497 6151607 6157932 6442551 6466936 6490594 6751617 6865541 6175837 5027316 5561798 5581750 5884327 5455944 5864849 5764897 6233585 5596754 5758356 5280615 6122642 4823304 5359730 5430719 5438509 5687372).pn. (5758069 5806074 5819066 5893912 5905860 5960424 6088757 6098879 6185650 6189007 6237005 6289343 6324683 6816860 6898587 5455914 5706463 4316245 4369494 4374409 4394725 4395757 6157927 6272675 6304873 6339772 6349310 6397227 6453313 6604102 5758351 4627019 4961134 5263154 5706511 6078949 6446092

☐	L37	5408653 5557792 5761660 4949251 5333314 5596710 5797138 5845117 4575816 6347361 6385694 6578160 5179698).pn. (5727203 5857180 5940827 6012094 6105147 5333303 5680610 5781910 5278982 6112987 4868744 5241675 5263156 5442779 5561795 5564050 5721918 5845292 5857204 5862327 5960436 6205465 6625602 5220665 5258982 5291594 5432926 5455946 5499367 5524239 5524241 5561794 5581691 5729733 5734896 5752026 5806076 5832517 5835915 5930794 5950212 5956713 5983225 6081810 6192377 6338146 6360228 6421688 6480847 H002064).pn. (H002064 6567928 6651073 6738971 6772363 6816873 5890154 6178440 5375232 6014674 6144941 6182186 6185663 6343339 6374267 6411969 6418438 5835764 5852732 5987557 6016143 6006229 5907673 5937409 6801919 4498145 5721905 5930800 5987463 6170063 5574902 6154847 5193188 5363121 5768587 5794241 5916307 6035301 6061708 6061708 6199055 6237001 6247023 6535997 6557082 6571270 5551046 5737601 5806075 5355477).pn. (5706500 5740353 6178463 4546240 6321306 6385702 6418514 6195709 5724581 5797026 6434544 4359630 4551799 4774662 4903200 5001624 5023774 5222224 5315511 5319766 5361267 5394536 5404470 5455406 5535340 5553266 5553258 5555382 5630055 5737547 5761486 5812799 5838995 5841369 5860110 5867735 5913021 5924120 5964838 6014649 6061772 6061772 6079030 6122692 6198985 6205363 6208906 6269427 6272515 6311254).pn.	295
☐	L36	(5440732 5870758 5287501 5778179 6219666 6219666 6298478 5944816 5452445 5485607 5561797 5878206 5953719 5940839 6772154 6766323 5717919 6253193 6363488 6389402 6427140 6016490 6308287 6457065 5465328 5201044 5317731 5504899 6138143 5263155 5497487 5535386 5799305 5832519 5864851 5872971 5893117 6353828 5440712 5903898 5878419 5734817 6009405 5920863 5890161 6219675 6219675 5530851 5875291 6148299)	880
☐	L35	L7 and L31	33
☐	L34	L6 and L31	4
☐	L33	L23 and L31	4
☐	L32	netscape.asn.	61
☐	L31	microsoft.asn.	4053
☐	L30	L29 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$).ti.	84
☐	L29	L28 not (ibm or (international adj1 business adj1 corporation))	406
☐	L28	L27 and (window or windows or menu$ or icon$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or internet or (world adj1 wide adj1 web) or www or hypertext or hyperlink$ or http or url$).ab.	503
☐	L27	navigat$.ab.	5135
☐	L26	((navigat$ near (window or windows or menu$ or icon$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or internet or (world adj1 wide adj1 web) or www or hypertext or hyperlink$ or http or url$)) not (ibm or (international adj1 business adj1 corporation)))	1388
		L24 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$	

☐	L25	or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$).ti.	91
☐	L24	hierarch$.ti.	1621
☐	L23	L22 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$).ti.	58
☐	L22	L21 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	1637
☐	L21	navigat$.ti.	3931
☐	L20	navigat$.ti. and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	1637
☐	L19	L18 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	24079
☐	L18	(hierarch$ or navigat$ or level$ or layer$ or node$ or branch$).ti.	61766
☐	L17	L7 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	3449
☐	L16	L6 and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	431
☐	L15	(L13 or L14) and (windows or menu$ or icon$ or folder$ or file$ or director$ or intefac$ or (web adj1 site$) or (web adj1 page$) or (internet adj1 site$) or (internet adj1 page$) or www or (world adj1 wide adj1 web) or internet or http or html or hypertext or hyperlink$ or url$)	22
☐	L14	(L6 or L7) and (rollback or rollbacktransaction or (rollback adj1 transaction) or rollback-transaction)	9
☐	L13	(L6 or L7) and (commit or committransaction or (commit adj1 transaction) or commit-transaction)	20
☐	L12	L11 and (hierarch$ or tree$ or level$ or layer$ or branch$ or node$)	95
☐	L11	L6 and (interfac$ or (world adj1 wide adj1 web) or internet or www or (internet adj1 site$) or (internet adj1 page$) or (web adj1 site$) or (web adj1 page$) or hyperlink$ or hypertext or html or http or url$)	131
☐	L10	L6 and ((interfac$ or (world adj1 wide adj1 web) or internet or www or (internet adj1 site$) or (internet adj1 page$) or (web adj1 site$) or (web adj1 page$) or hyperlink$ or hypertext or html or http or url$) near nest$)	7
☐	L9	L8 and (hierarch$ or tree$ or level$ or layer$ or branch$ or node$)	39

☐	L8	L6 and ((microsoft near windows) or windows or menu$ or icon$ or folder$)	56
☐	L7	nest$.ab.	9214
☐	L6	nest$.ti.	1899
☐	L5	((interfac$ or (world adj1 wide adj1 web) or internet or www or (internet adj1 site$) or (internet adj1 page$) or (web adj1 site$) or (web adj1 page$) or hyperlink$ or hypertext or html or http or url$) near nest$)	122
☐	L4	L3 and nest$.ab.	1
☐	L3	L2 and nest$.ti.	1
☐	L2	L1 and (hierarch$ or tree$ or level$ or layer$ or branch$ or node$)	144
☐	L1	((microsoft near windows or windows or menu$ or icon$ or folder$) near nest$)	174

END OF SEARCH HISTORY